ACADEMY LIVE

HORNBILL

04.09.2024

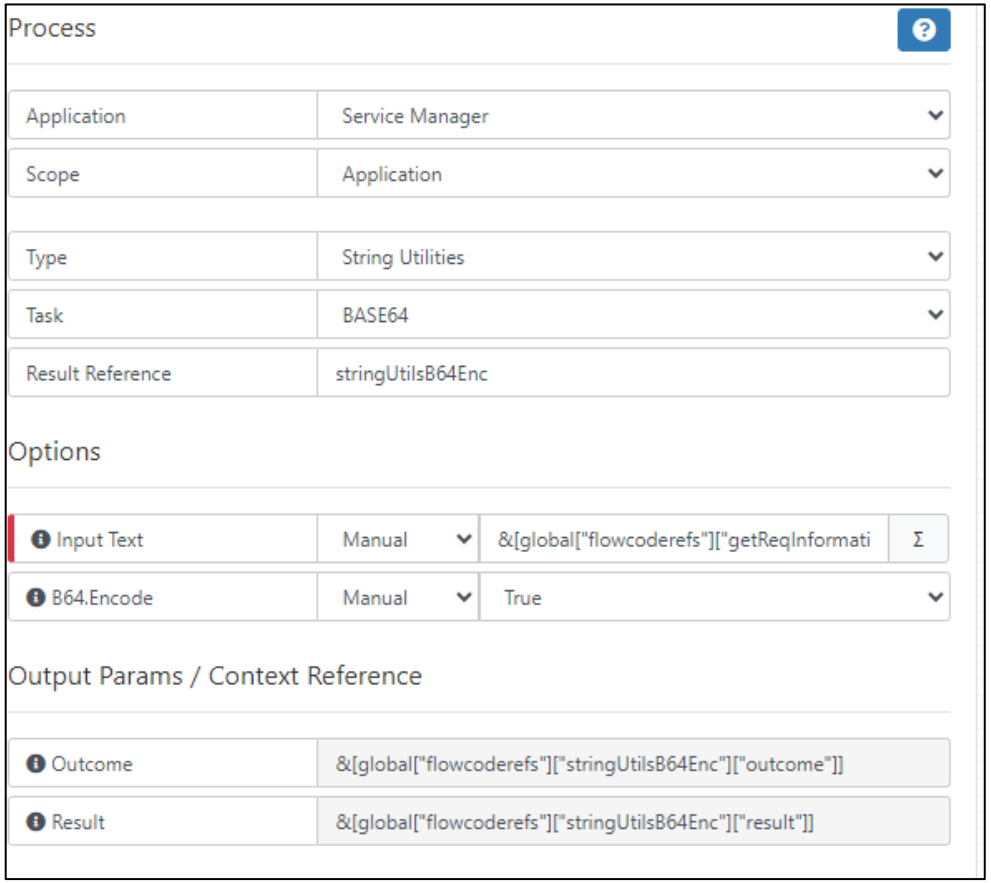
Workflow Utilities

Strings

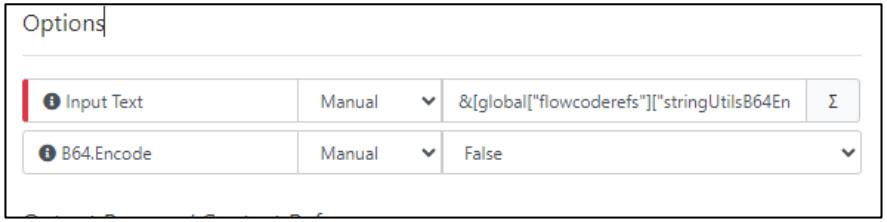
Base64 Encode/Decode

- **Description**: Converts a string to and from Base64 encoding. This is useful when transmitting binary data via APIs or embedding data within URLs. While it's not encryption, it obfuscates the data to make it less obvious.
- **Use Case**: When making API calls that send sensitive information (like image files or credentials), Base64 encoding can help transmit the data safely. For example, if you're passing a user profile picture in a workflow, you could encode it before sending it in an API request.
- **Example**: You have a PDF document that needs to be sent to another service via an API. You would encode it in Base64, include it in the API request, and decode it on the receiving end.

Encode



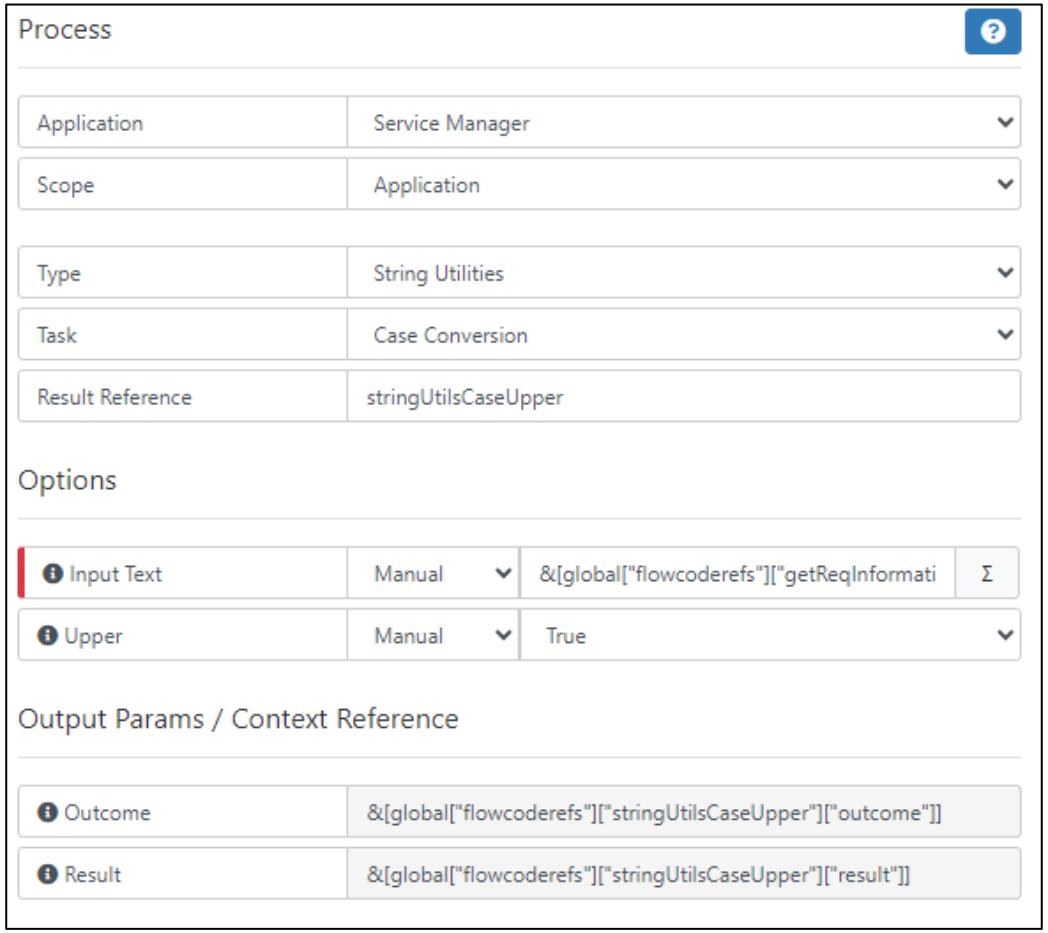
Decode



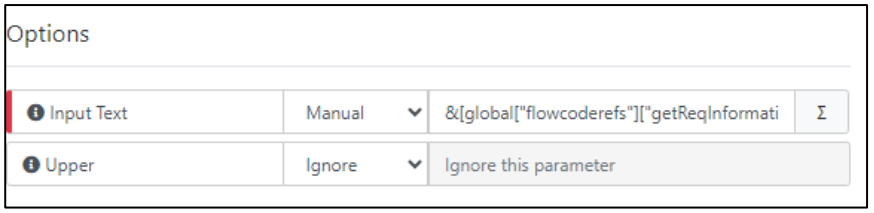
Case Conversion

- **Upper**: Converts all characters in a string to uppercase.
- **Lower**: Converts all characters to lowercase.
- **Use Case**: This is helpful when ensuring consistency in data input. For instance, if you want to ensure all usernames are in lowercase before storing them in a database.
- **Example**: You receive a list of names for request assignment, but some are in mixed case. You can convert them all to uppercase or lowercase to avoid issues in comparisons or database lookups.

Upper

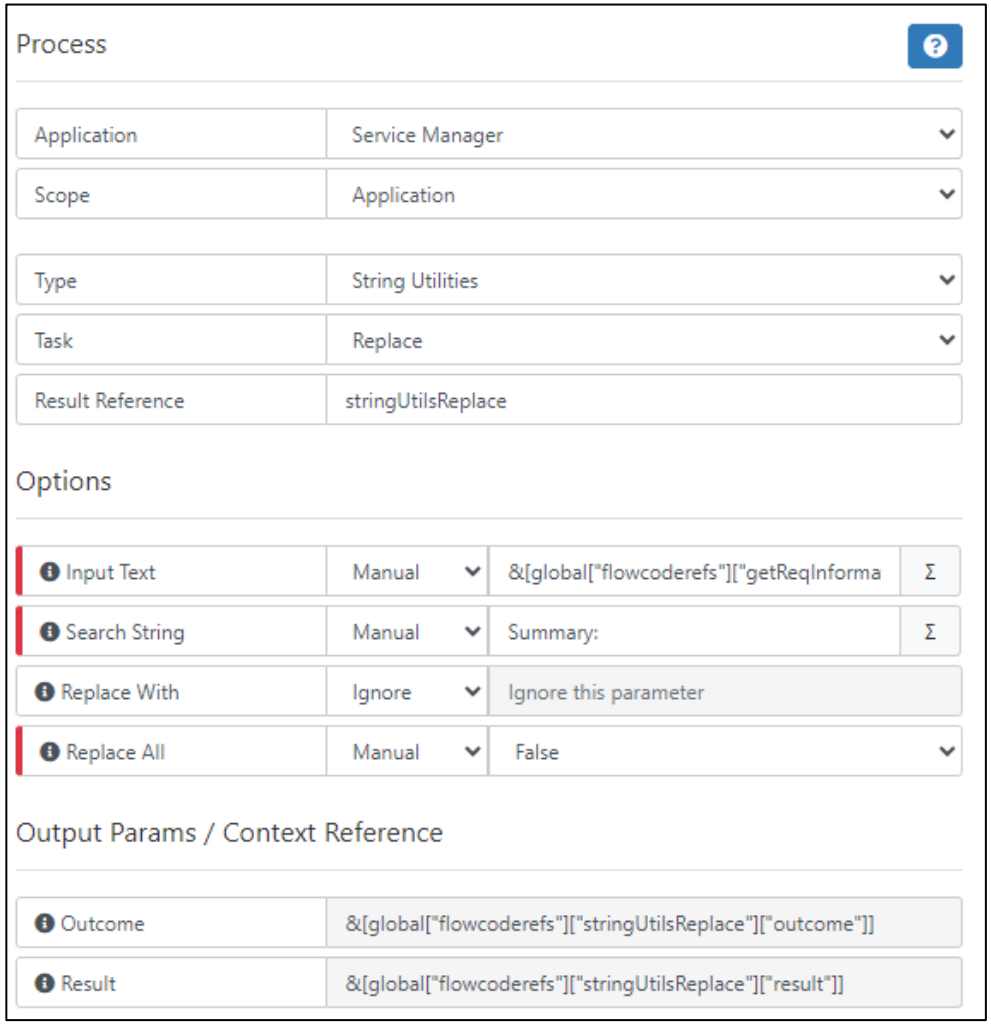


Lower



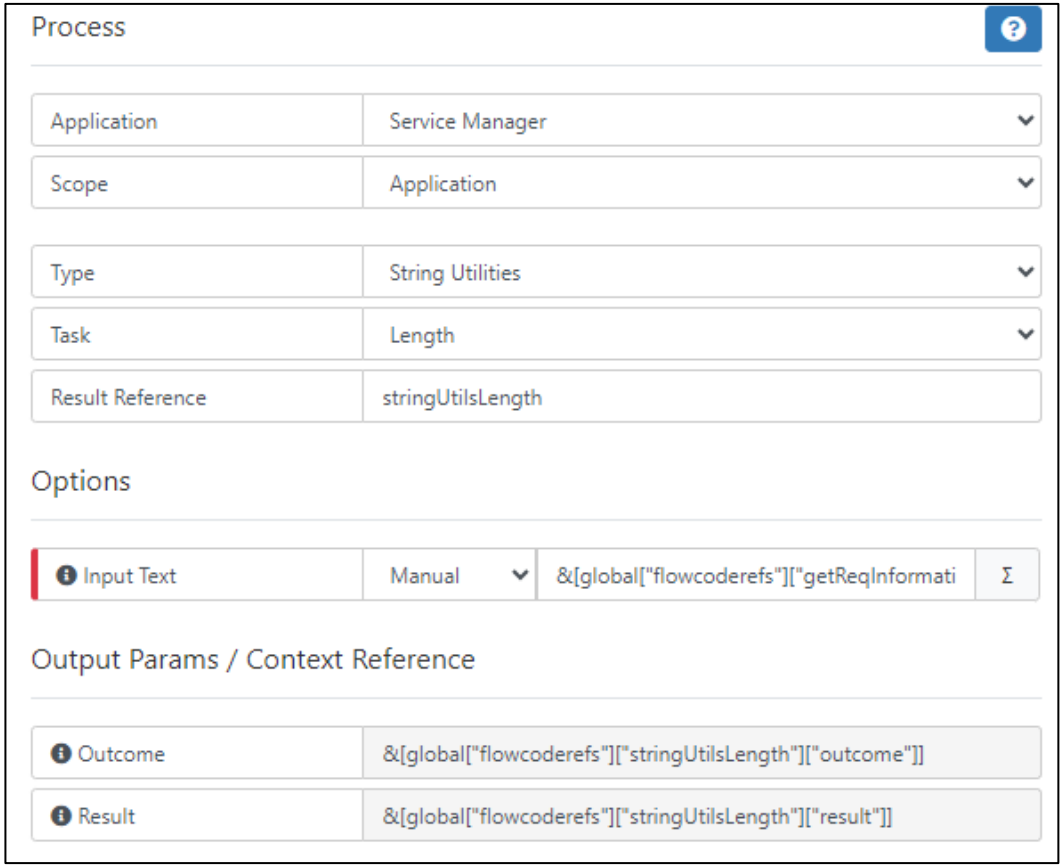
Replace

- **Description**: Replaces a specified part of the string with another string. This can be done for either the first occurrence or all matches.
- **Use Case**: Useful for cleansing data. For instance, if you're processing email addresses and some contain incorrect characters, you can use "Replace" to fix those.
- **Example**: If an email field accidentally contains a space, "Replace" can remove all spaces and replace them with the correct punctuation.



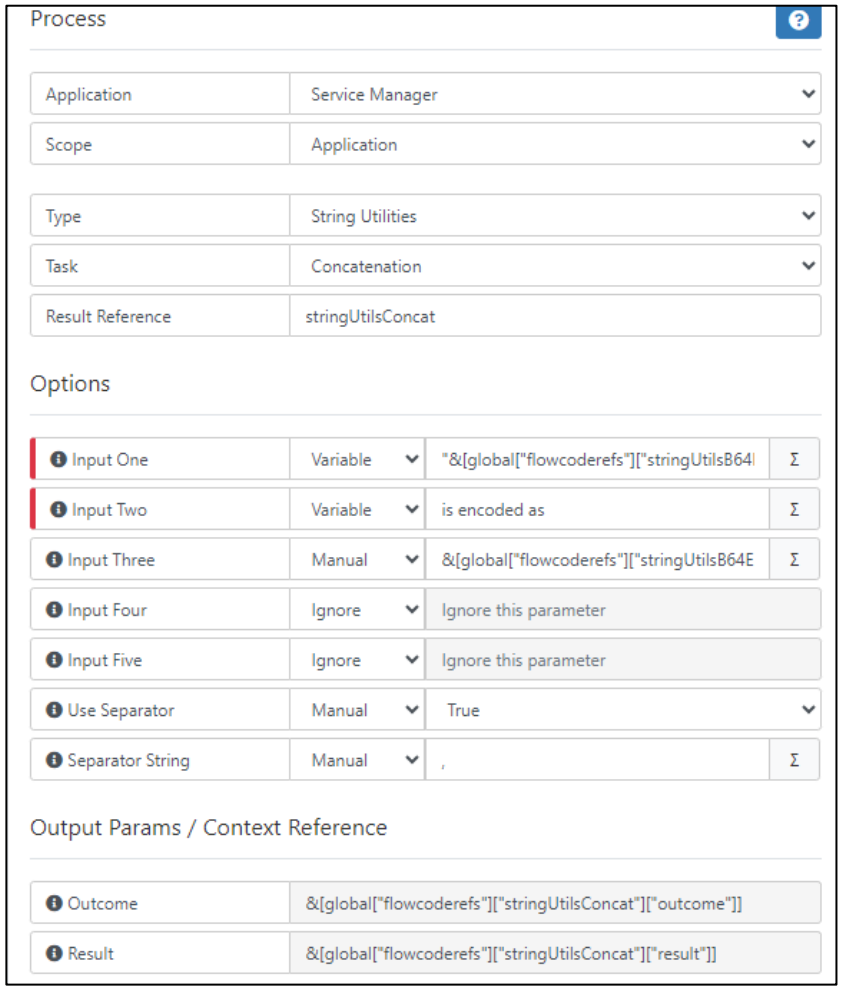
Length

- **Description**: Returns the length of a string.
- **Use Case**: You can use this to validate the length of inputs like passwords or IDs before performing further actions.
- **Example**: In a workflow, you might ensure that a request number is exactly 8 characters long. If it's shorter or longer, the process can raise an error.



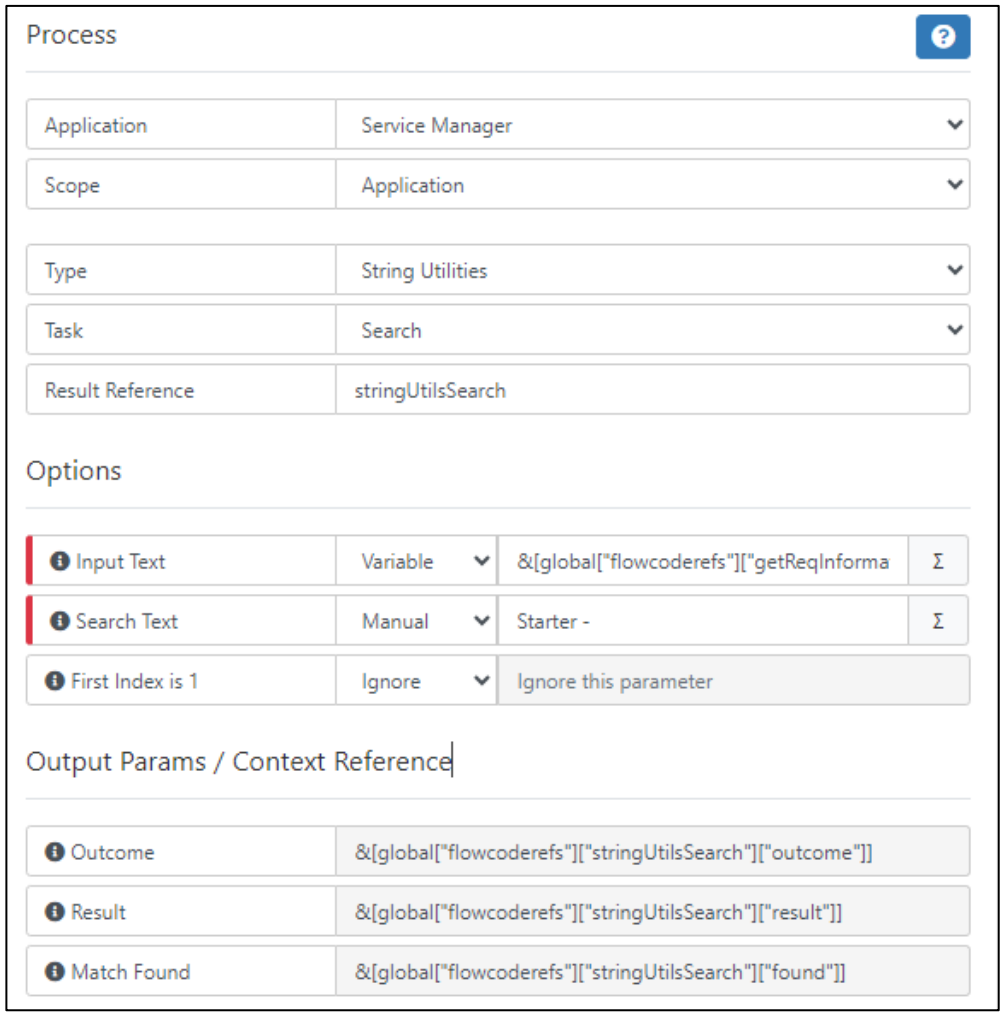
Concatenation

- **Description**: Combines up to 5 strings with an optional separator, which is often used to combine multiple pieces of information into one string.
- **Use Case**: When populating email fields or creating log entries, concatenation helps in bringing together multiple data points into one entry.
- **Example**: You're sending a notification email and need to include both the user's name and the request number. You concatenate these fields with a separator like a comma to create a cohesive message.



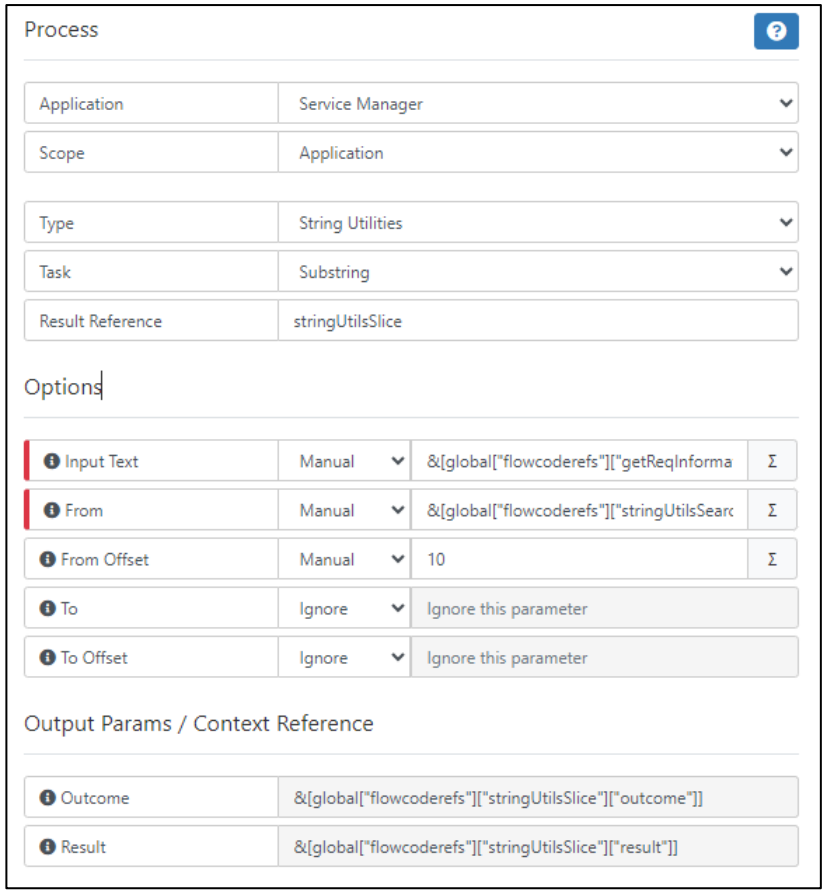
Search

- **Description**: Searches for a specific substring within a string and returns its starting position. This is used when you need to check if a string contains a particular value.
- **Use Case**: Use this to verify whether an email contains a particular domain before performing any further actions, like sending an email.
- **Example**: In a workflow, if you're checking to see if an email address contains "@company.com", you can use the "Search" utility to ensure the input matches the desired domain.



Substring

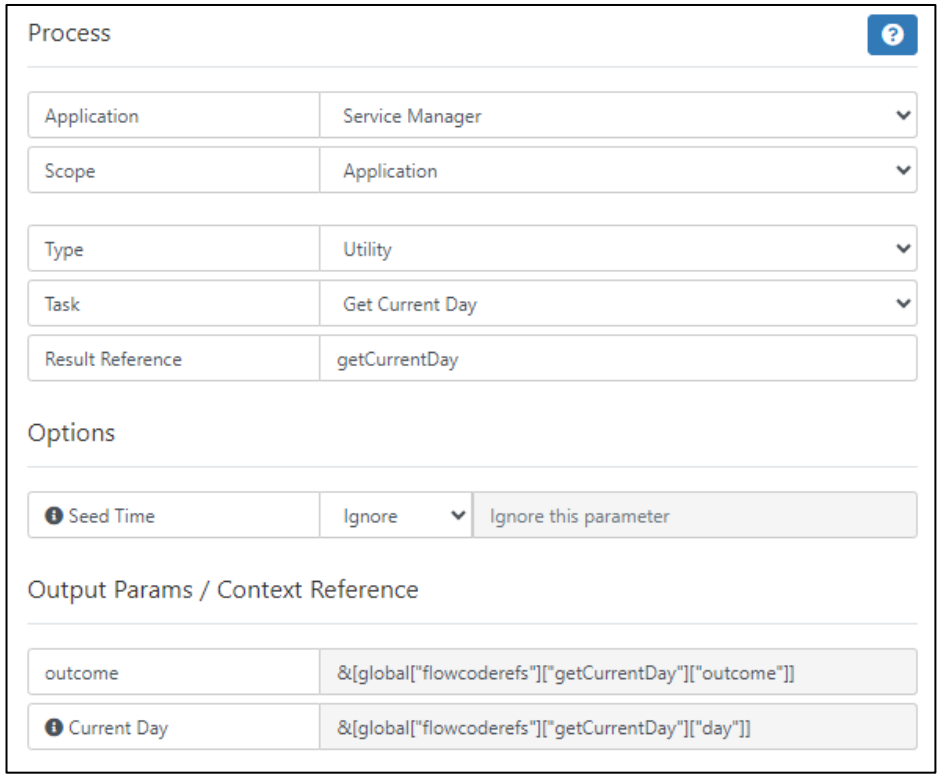
- **Description**: Extracts a portion of a string from a specified starting point.
- **Use Case**: This is often used when pulling specific parts of a string, such as an ID or a username from an email address.
- **Example**: You receive an input in the format "john.doe@company.com" and need to extract the username part ("john.doe") for use in another step.



Dates

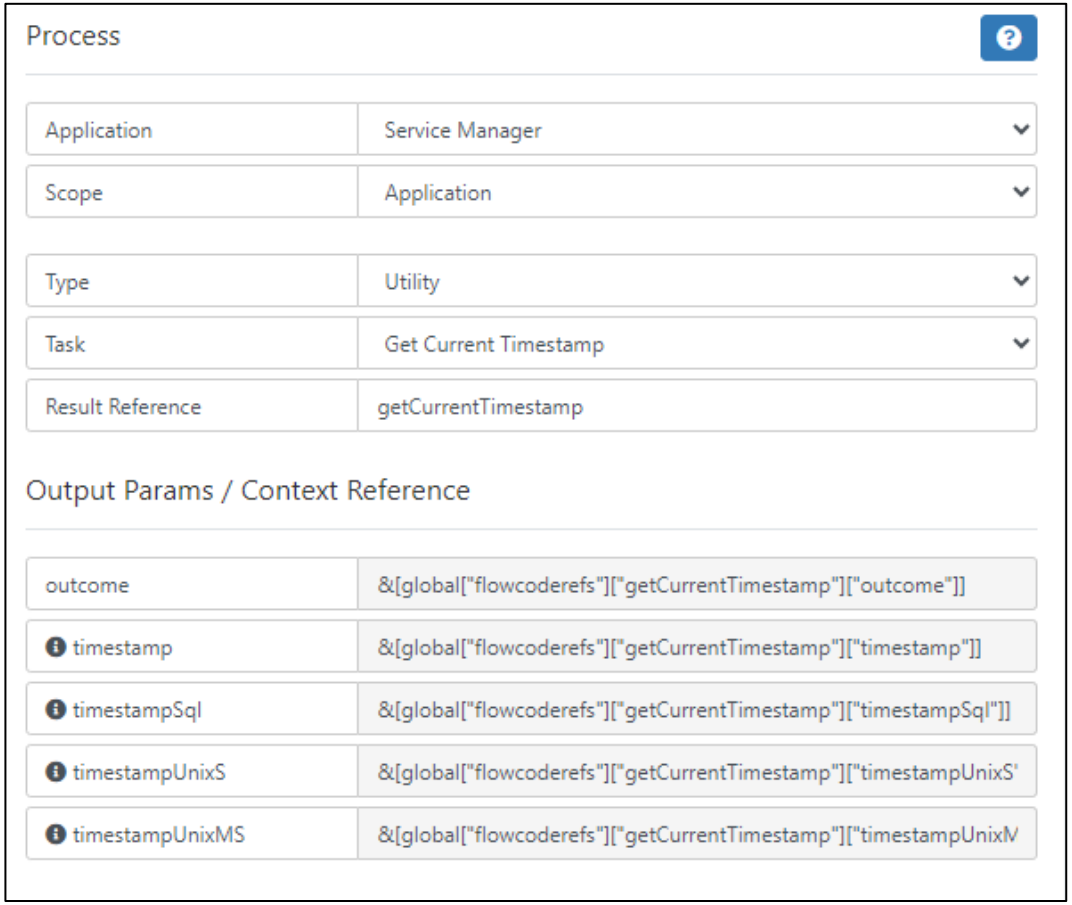
Get Current Day

- **Description**: Retrieves the current day (e.g., Monday, Tuesday).
- **Use Case**: This is useful when automating workflows based on the day of the week, such as sending reminders only on business days.
- **Example**: If you have a task that should only be triggered on weekdays, this utility can check whether today is a weekday before proceeding with the action.



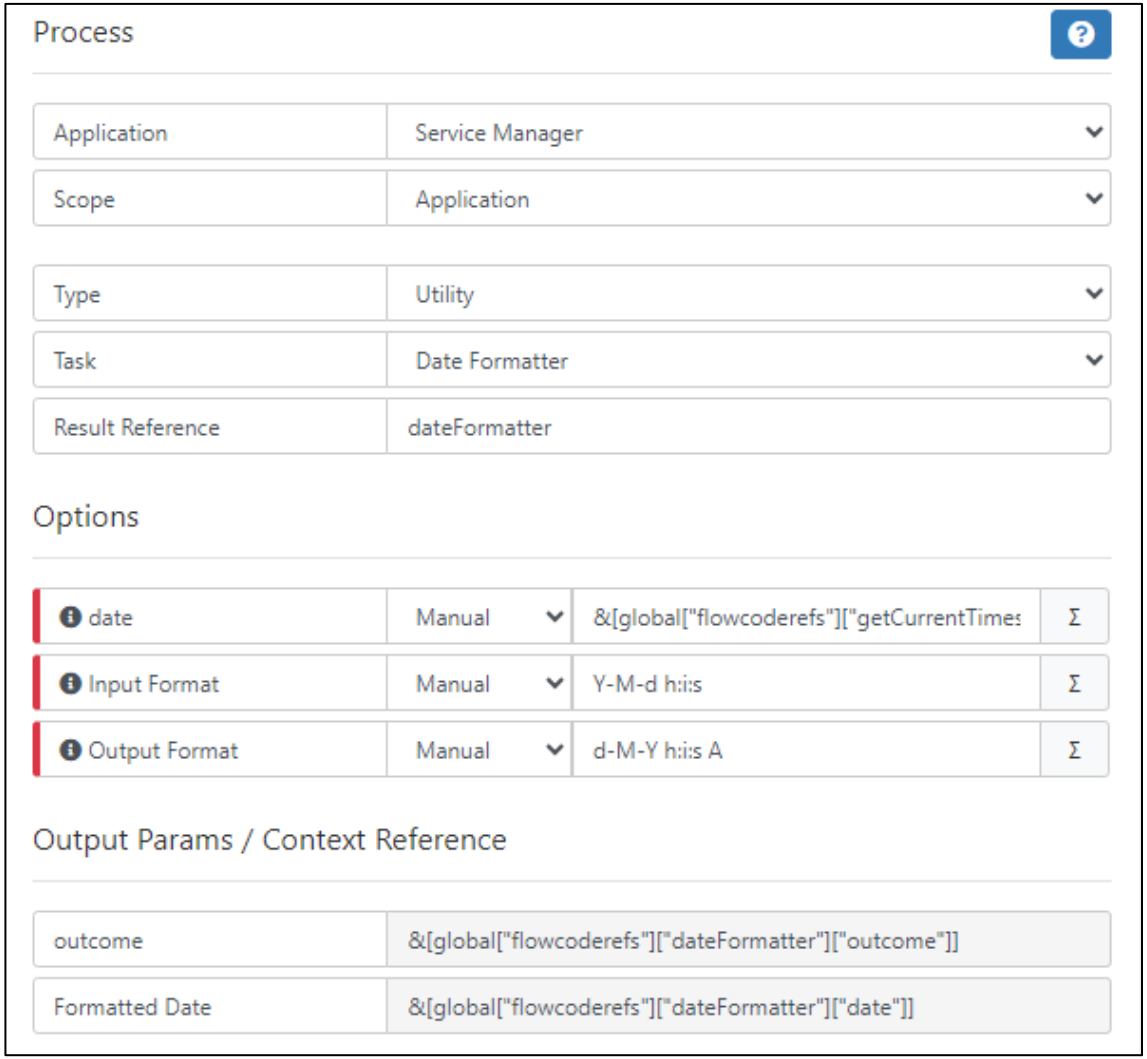
Get Current Timestamp

- **Description**: Returns the current date and time in several formats:
  - **Timestamp**: 2024-08-29 06:56:10Z (ISO 8601 format)
  - **SQL format**: 2024-08-29 06:56:11
  - **Milliseconds/Seconds**: Number of milliseconds/seconds since 1970 (Unix Epoch)
- **Use Case**: Use this to timestamp log entries, set deadlines, or calculate durations. This can also be helpful when comparing timestamps for workflow actions.
- **Example**: When logging ticket changes, you can record the exact time of the update in both human-readable (ISO) and Unix timestamp formats for further reporting.



Date Formatter

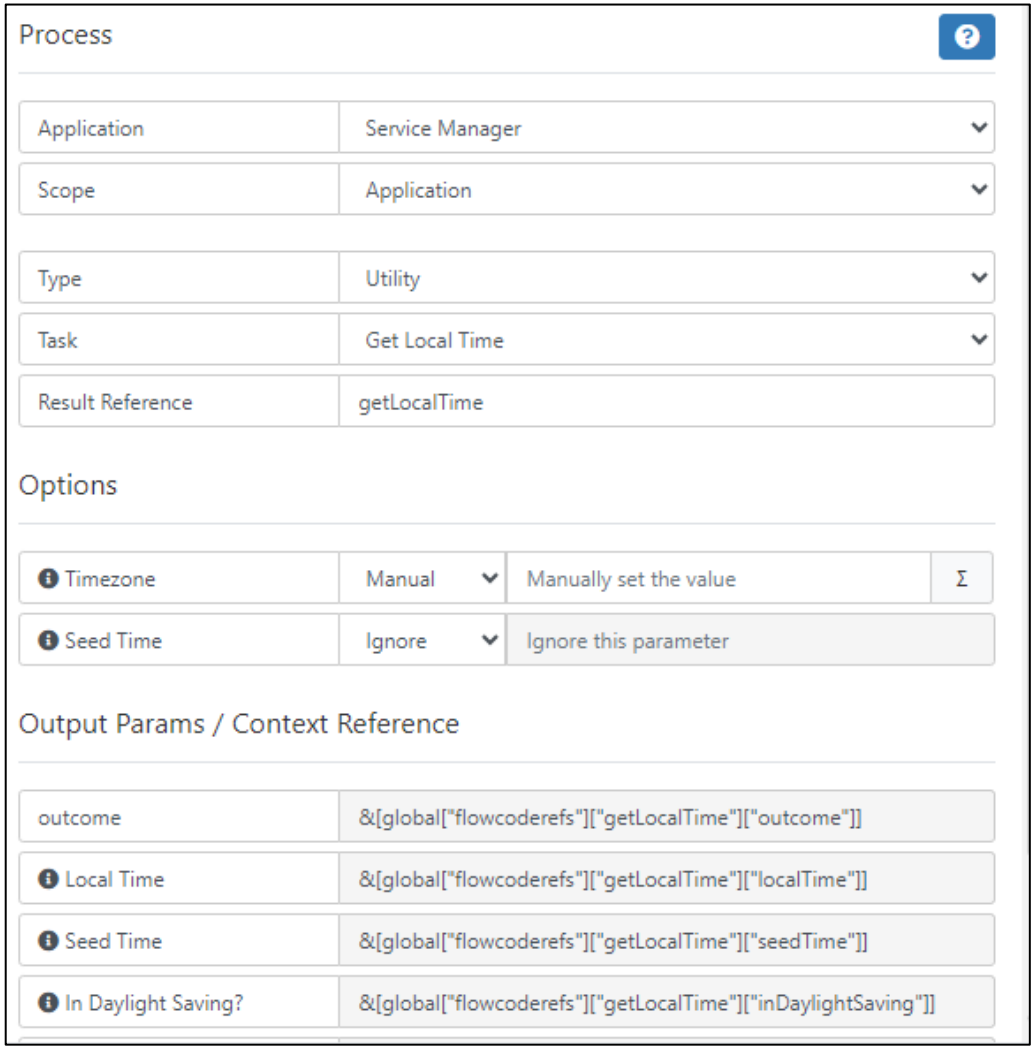
- **Description**: Converts dates from one format to another, using ISO 8601 strings.
- **Use Case**: When you need to adjust date formats to suit regional preferences, such as converting a date from YYYY-MM-DD to DD/MM/YYYY.
- **Example**: If you're working with teams across different time zones and regions, this utility helps you display dates in the appropriate local format (e.g., UK vs. US).



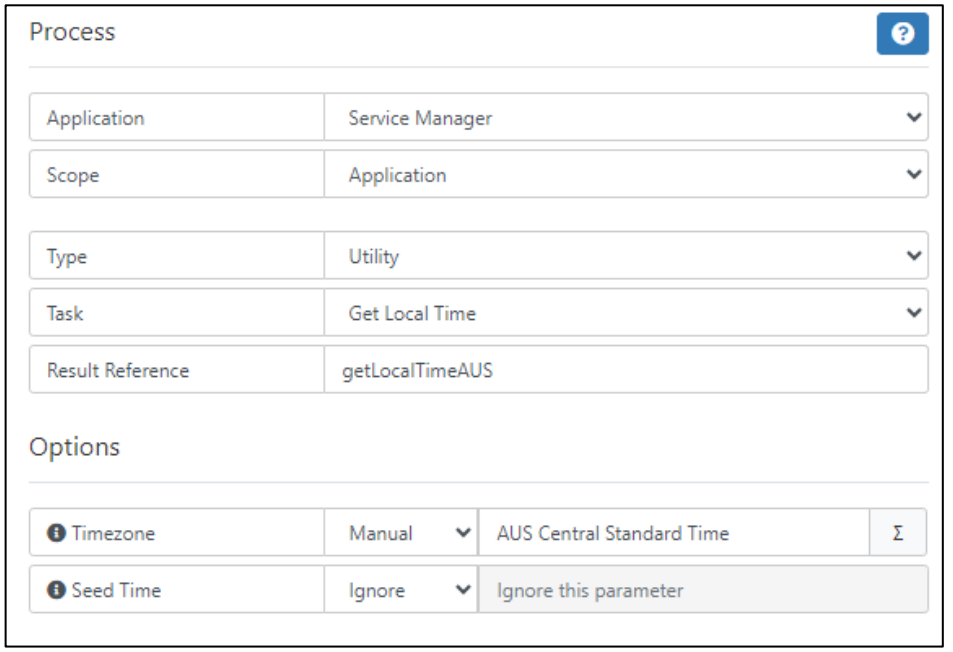
Get Local Time

- **Description**: Retrieves the local time based on the current session time zone or a specified time zone.
- **Use Case**: Use this when dealing with international users or data centers, ensuring that operations happen in the right time zone.
- **Example**: If you're working with teams in different regions (e.g., London vs. Sydney), you can fetch and display the appropriate time zone for notifications or scheduled actions.

Retrieve the local time based on the current session

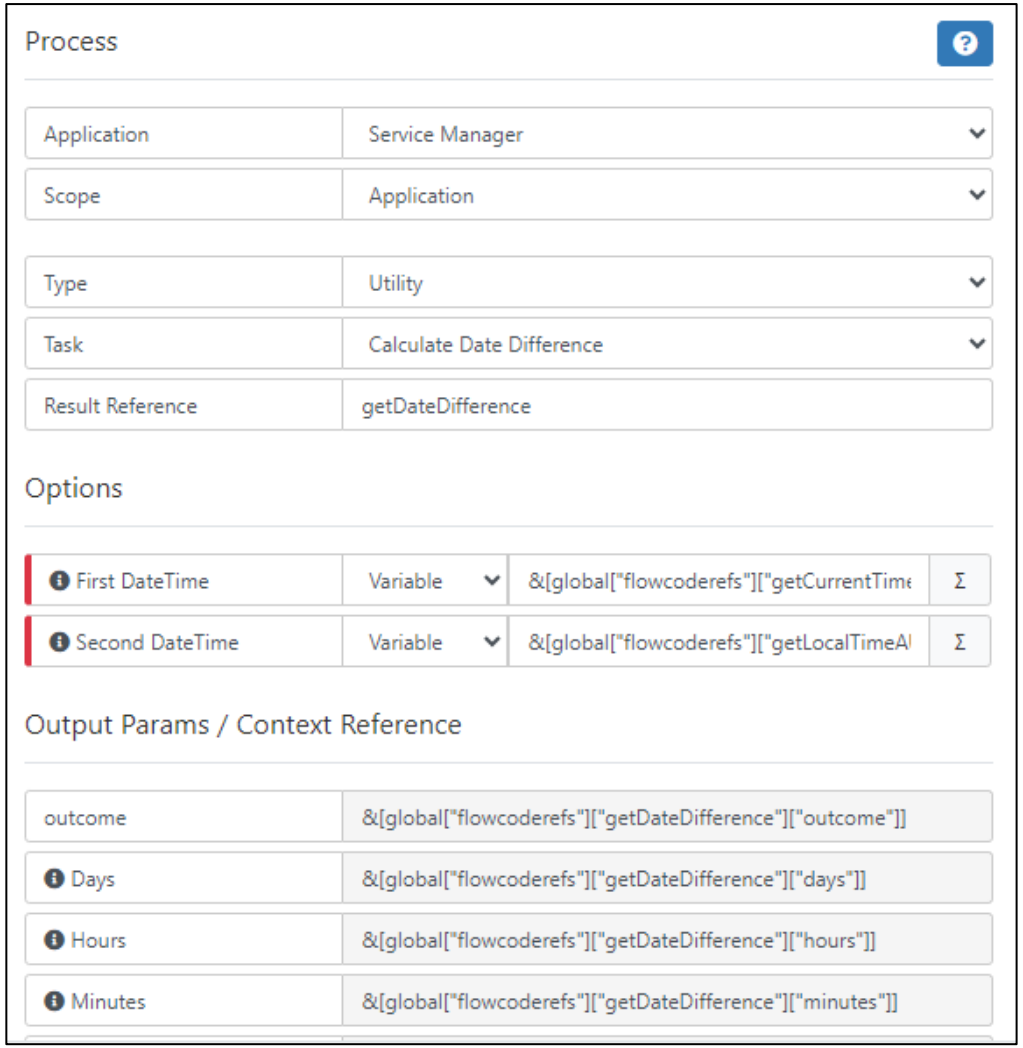


Retrieves the Local time using the Aus Central Standard Time



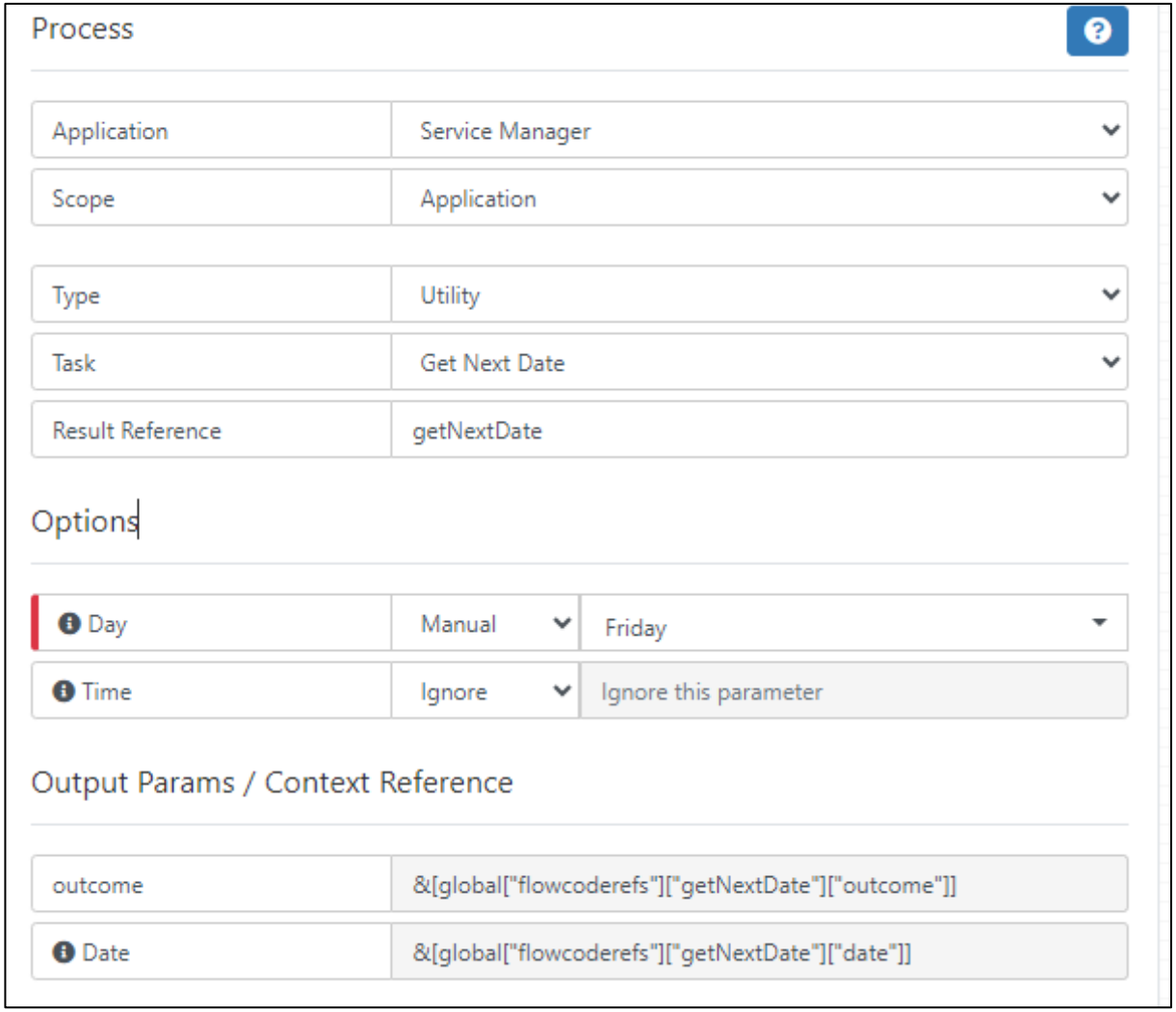
Calculate Date Difference

- **Description**: Computes the difference between two dates or times in seconds, minutes, hours, or days.
- **Use Case**: Helpful for calculating SLAs or the time it takes to resolve an issue.
- **Example**: When calculating the time elapsed between when a ticket was created and when it was resolved, this utility returns the exact difference, which can be used for reporting and compliance.



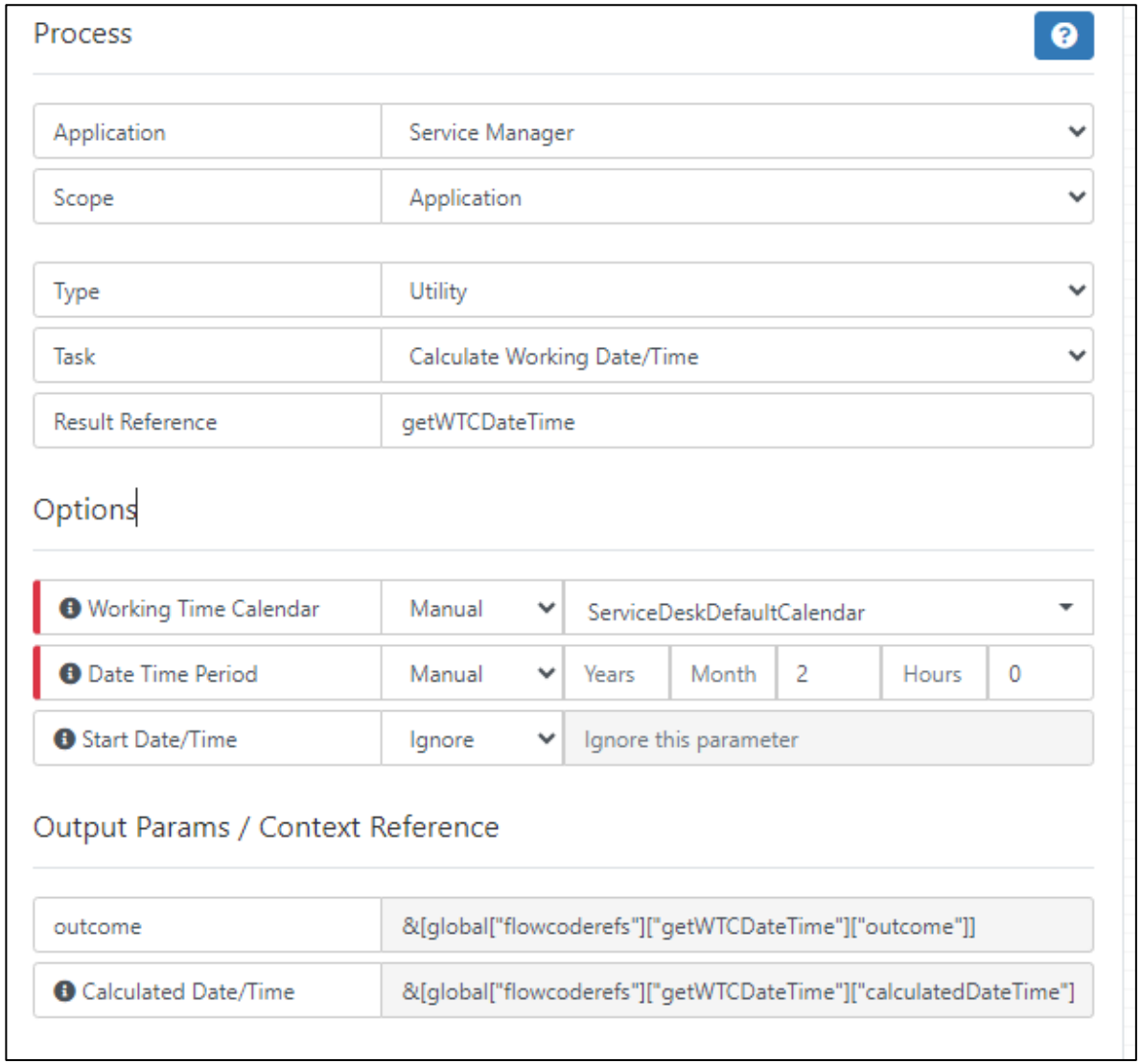
Get Next Date

- **Description**: Retrieves the next date for a specified day of the week.
- **Use Case**: Used for scheduling recurring events, such as weekly status reports or maintenance windows.
- **Example**: In a weekly maintenance window that happens every Monday, this utility can automatically calculate the next Monday to set the next window.



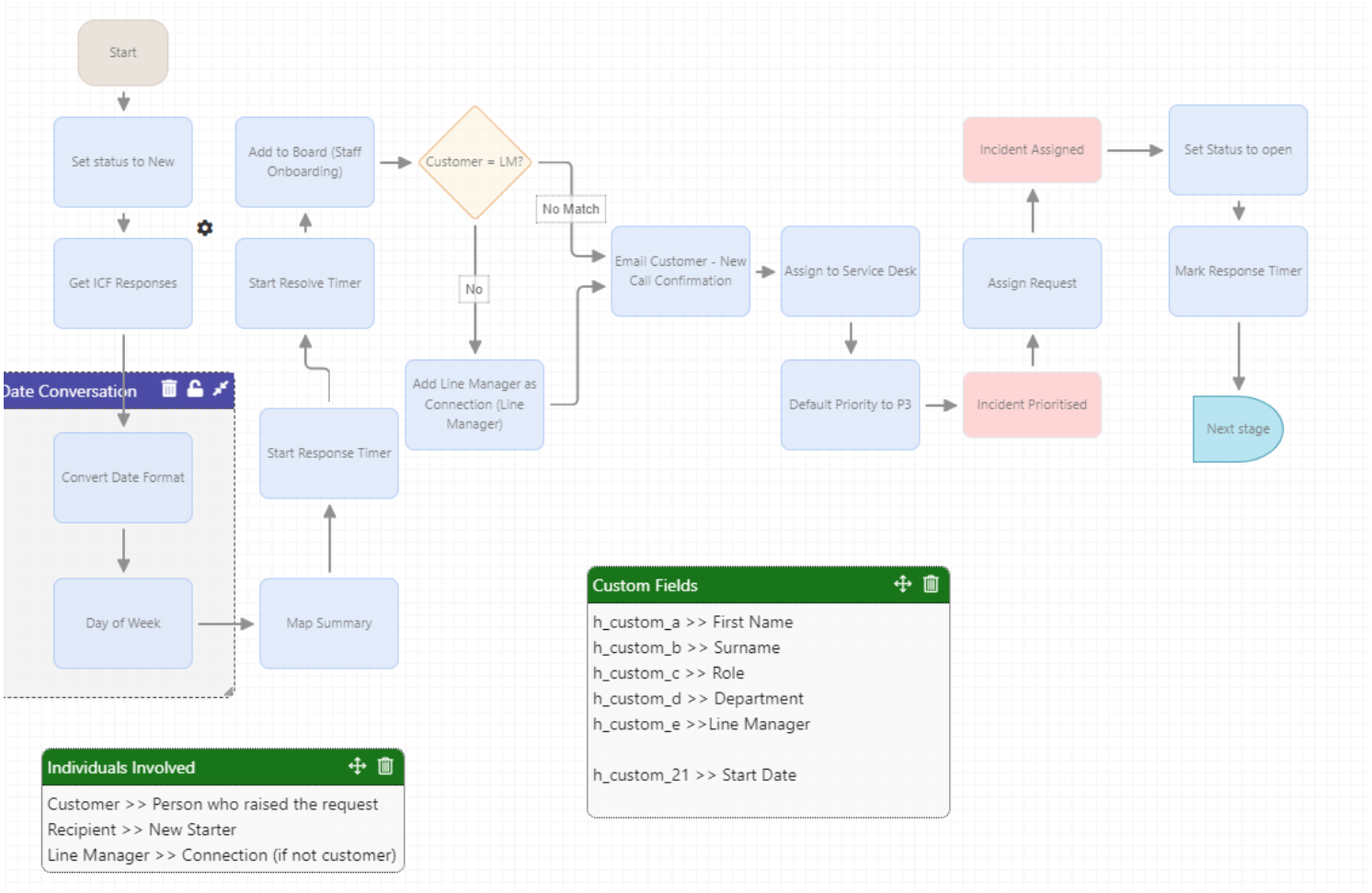
Calculate Working Date/Time

- **Description**: Calculates a future date based on a provided duration, considering the configured working hours.
- **Use Case**: This is essential when suspending tasks based on working hours, such as waiting for a customer reply during business hours.
- **Example**: If you're waiting for a customer's response within 2 working days, this utility accounts for weekends or holidays to give you the accurate due date.

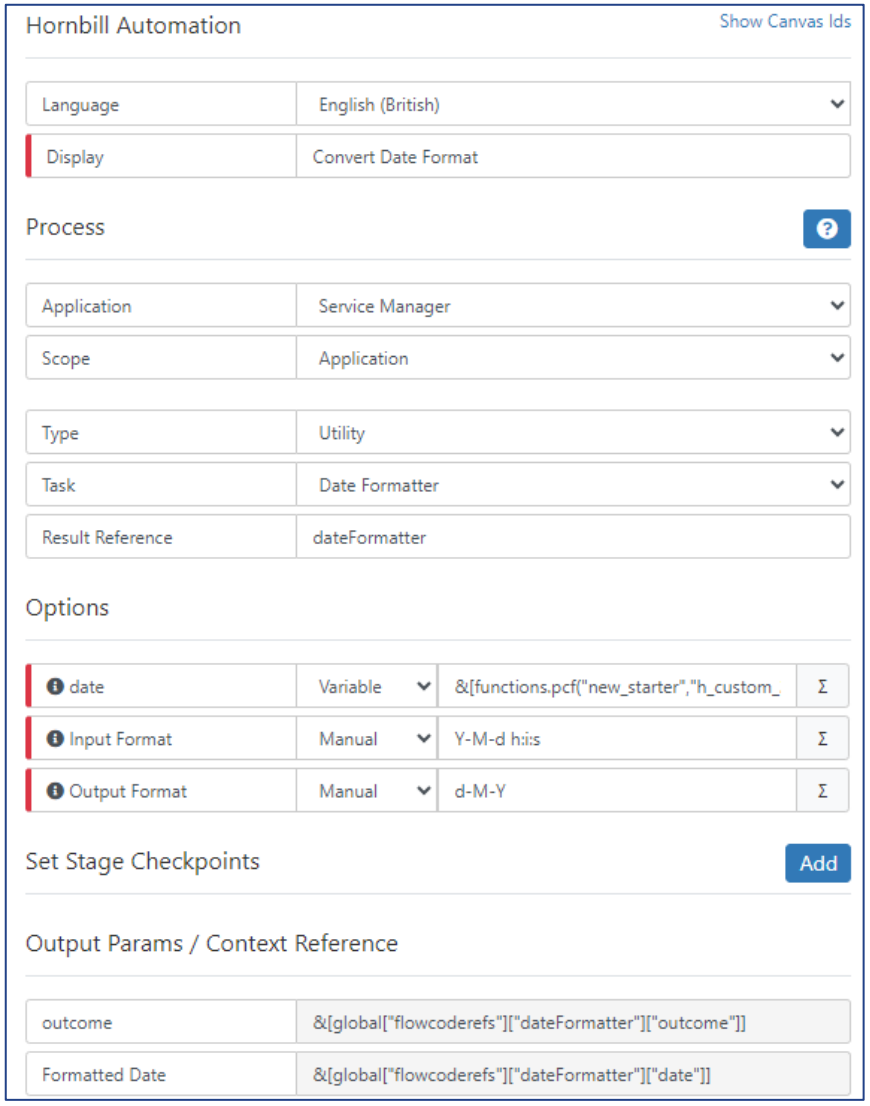


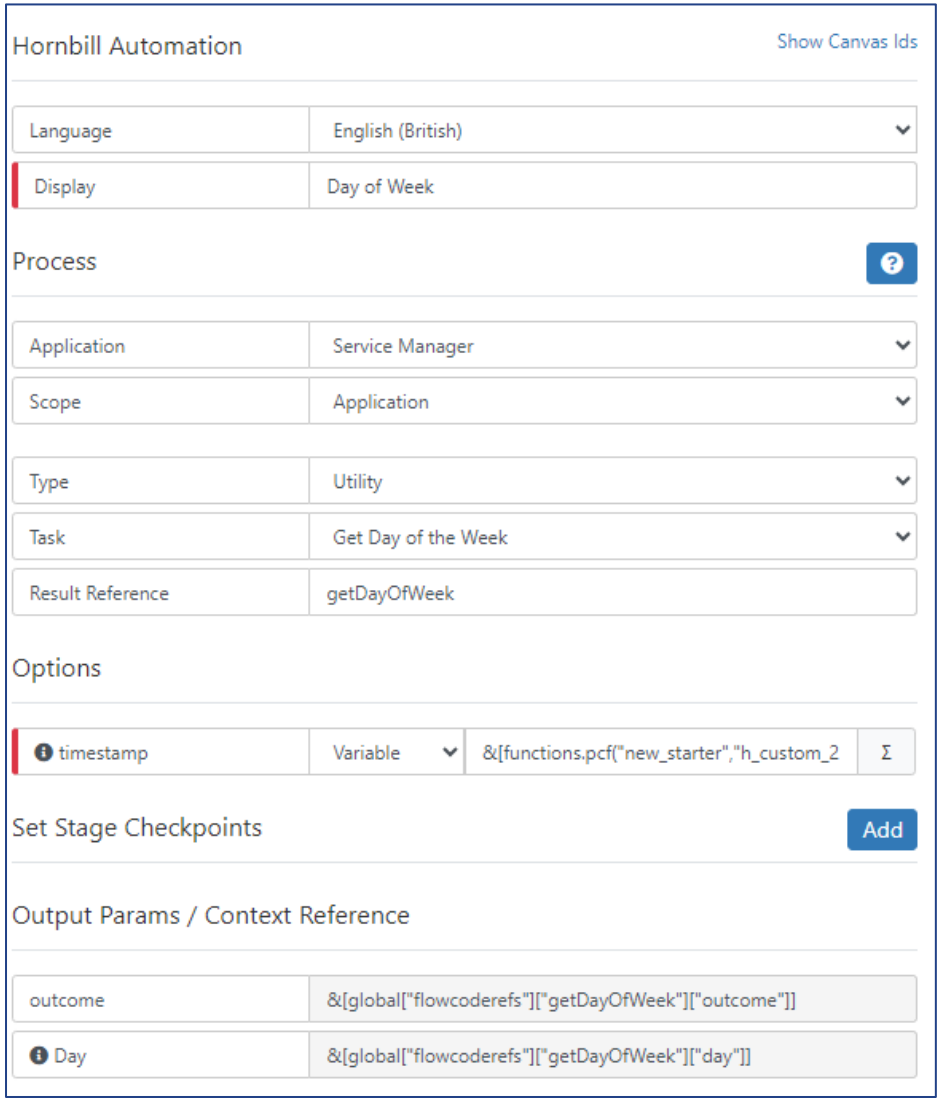
Example of using a Workflow utility

The following activity utilises two nodes to be inserted into the New Starter Workflow. The changes will manipulate data provided during the Intelligent Capture to add the start in the request summary using the following format**: - Start Date: Thursday 26-Sep-2024**



The following two nodes will be added:





The variables used should be correct and map to those used within the current process, however, I am not sure which Intelligent Capture you will be using, so may need checking.

The existing Map Summary node will need the Summary Option modified to the following:

New Starter: &[functions.pcf("new_starter","h_custom_a")] &[functions.pcf("new_starter","h_custom_b")] - Start Date; &[global["flowcoderefs"]["getDayOfWeek"]["day"]] &[global["flowcoderefs"]["dateFormatter"]["date"]]